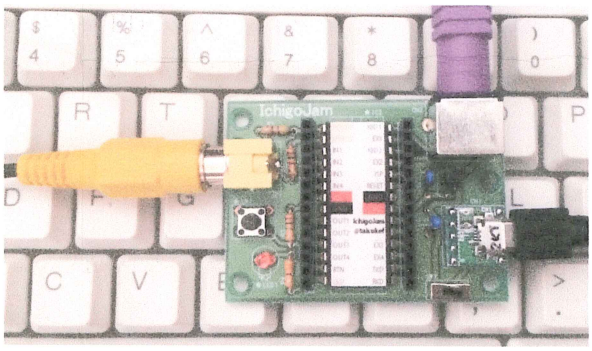
こどもパソコン「IchigoJam」 プリント基板キット

IchigoJam は、PS/2 キーボード、ビデオケーブルと接続するテレビ、マイクロ USB ケーブル（電源）を接続するだけで、ゲームやツール、ロボット制御などを BASIC でプログラミングできるコンピューターです。



・　部品一覧（組み立て推奨順）

IchigoJam プリント基板

R1 抵抗 470Ω 黄紫茶金

R2 抵抗 100Ω 茶黒茶金

R3 抵抗 1MΩ 茶黒緑金

R4 抵抗 330Ω 橙橙茶金

IC2 マイコン LPC1114（IchigoJam Core） ※向きあり、切り込みがある方を上に

CN1 ピンヘッダ（7 ピン）5 ピンと、1 ピン 2 つにペンチで分けて差し込む

CN1 マイクロ USB 端子 ※のせて上からハンダ付け、次に下からハンダ付け

C1, C2 コンデンサ 0.1μF、2 つ

IC1 3 端子レギュレーター ※向きあり、平たい方が内側

LED1 赤色 LED ※向きあり、足が長いほうが内側

SW1 スライドスイッチ ※向きなし、右側にスライドしておく（OFF の位置）

SW2 タクトスイッチ（ボタン）※足が開く方向が左右になるように

CN3, CN4 ピンソケット（14 ピン）、2 つ

CN2 PS/2 端子 ※1 箇所つけた後、プリント基板に押し付けるときれいに仕上がる

JC1 ビデオ端子

・　動かし方

キーボード、テレビ、電源をつなぎ、スライドスイッチを左に動かすと起動します。

LED 1 （リターン／エンター）と打ち込むと LED が点灯。LED 0 で消灯。

その他、リファレンス、入門、サポートなど http://ichigojam.net/ をご参照ください

2014/09/28 こどもパソコン IchigoJam <ichigojam.net@gmail.com>

# IchigoJam BASIC リファレンス - 0.9.7

### IchigoJamとは？

IchigoJam（いちごジャム）は、PS/2キーボード、ビデオケーブルと接続するテレビ、マイクロUSBケーブル（電源）を接続するだけで、ゲームやツール、ロボット制御などをBASICでプログラミングできるコンピューターです。スライドスイッチを左に動かすと起動します。

LED 1（エンター）と打ち込むとLEDが点灯します。カーソルを動かし LED 1 を LED 0 に書き換えてエンターで消灯します。その他、入門、サンプル、サポートなど http://ichigojam.net/ をご参照ください。

### 基本操作

| 操作 | 解説 | ver |
|---|---|---|
| キー | 文字を入力する | |
| Shift | キーと共に押し記号などを入力する | |
| Enter | コマンドを実行する（プログラム変更時もその行でEnter） | |
| カーソルキー | カーソルキーを移動する | |
| Backspace | 1文字前を消す | |
| Delete | 1文字先を消す | |
| CapsLock | 大文字と小文字を切り替える | |
| Insert | 上書きモード、挿入モードを切り替える | |
| カタカナ/右ALT | アルファベットとカタカナ（ローマ字入力）を切り替える | 0.8.2 |
| ESC | プログラムを中断する | |
| ファンクションキー | F1:CLS、F2:LOAD、F3:SAVE、F4:LIST、F5:RUN、F6:?FREE()、F7:OUT0、F8:VIDEO1 | |

### 初級コマンド

| コマンド | 解説 | 例 | ver |
|---|---|---|---|
| LED 数 | 数が1なら光り、0なら消える | LED 1 | |
| WAIT 数 | 数の数値フレーム分待つ 60で約1秒 | WAIT 60 | |
| : | コマンドを連結する | WAIT60:LED1 | |
| 行番号 コマンド | プログラムとしてコマンドを記録する | 10 LED1 | |
| 行番号 | その行番号のプログラムを消す | 10 | |
| GOTO 行番号 | 次に実行する行を決める ver0.9.7以降は数でもOK | GOTO 10 | |
| PRINT 数や文字列 | 文字を表示する ";" で連結できる（文字列は"で囲む） 省略形：? | PRINT "HI!" | |
| RND(数) | 0から数未満の正数をランダムに返す | PRINT RND(6) | 0.3 |
| CLS | 画面を全部消す | CLS | |
| CLV | 変数、配列を全部0にする | CLV | 0.9.4 |
| INPUT (文字列,)変数 | キーボードからの入力で数値を変数にいれる | INPUT "ANS?",A | 0.6 |
| LET 変数,数 | アルファベット1文字を変数として数の値を入れる 省略形：変数=数 | LET A,1 | 0.7.6 |
| LOCATE 数,数 | 次に文字を書く位置を横、縦の順に指定する、縦=-1で無表示 省略形：LC | LOCATE 3,3 | |
| VPEEK(数,数) | 画面上の指定した位置に書かれた文字コードを所得する、指定なしで現在位置 省略形：SCR | PRINT VPEEK(0,0) | 0.7.2 |
| BTN() | ボタンが押されていれば1 | PRINT BTN() | 0.6 |
| INKEY() | キーボードから1文字入力する（入力がないときは0） | PRINT INKEY() | |
| CHR$(数) | 文字コードに対応する文字を返す コンマ区切りで連続表記できる(ver0.9.4以降) | PRINT CHR$(65) | 0.7.4 |
| ASC("文字") | 文字に対する文字コードを返す | PRINT ASC("A") | 0.8.7 |
| SCROLL 数 | 0:上、1:右、2:下、2:左 方向に1キャラクター分スクロール | SCROLL 2 | 0.9.7 |
| OUT 数,数 | OUT1-6に0または1を出力する(5,6は0.9.0以降) | OUT 1,1 | 0.8.7 |
| IN(数) | IN1-4から入力する（0または1） | LET A,IN(1) | 0.8.7 |
| ANA() | 外部入力IN2の電圧(0V-3.3V)を0-1023の数値で返す | A=ANA() | 0.8.4 |
| TICK() | 時間を返す（1/60秒で1進む） | PRINT TICK() | 0.8.1 |
| CLT | 時間をリセットする | CLT | 0.8.0 |
| 数 + 数 | 足し算する | PRINT 1+1 | |
| 数 - 数 | 引き算する | PRINT 2-1 | |
| 数 * 数 | 掛け算する | PRINT 7*8 | |
| 数 / 数 | 割り算する（小数点以下は切り捨てられます） | PRINT 9/3 | |
| 数 % 数 | 割り算した余りを返す | PRINT 10%3 | |
| (数) | カッコ内は優先して計算する | PRINT 1+(1*2) | |
| IF 数 THEN 次 ELSE 次2 | 数が0でなければ次を実行する THENとELSEは省略可 ELSEはver0.9.4以降 | IF A=B LED 1 | |
| 数 = 数 | 比較して等しい時に1を返す(==でも可) | IF A=B LED 1 | |
| 数 <> 数 | 比較して等しくない時に1を返す(!=でも可) | IF A<>B LED 1 | |
| 数 <= 数 | 比較して以下の時に1を返す | IF A<=B LED 1 | |
| 数 < 数 | 比較して未満の時に1を返す | IF A<B LED 1 | |
| 数 >= 数 | 比較して以上の時に1を返す | IF A>=B LED 1 | |
| 数 > 数 | 比較してより大きい時に1を返す | IF A>B LED 1 | |
| 式 AND 式 | 両方の式が正しい時に1を返す | IF A=1 AND B=1 LED 1 | |
| 式 OR 式 | どちらかの式が正しい時にに1を返す | IF A=1 OR B=1 LED 1 | |
| RUN | プログラムを実行する | RUN | |
| LIST | プログラムを表示する | LIST | |
| NEW | プログラムを全部消す | NEW | |
| LOAD 数 | プログラムを読み出す（0～2の3つ、省略すると0 / 100-226 外付けEEPROM） | LOAD | 0.8.0 |
| SAVE 数 | プログラムを保存する（0～2の3つ保存できます、省略で0 / 100-226 外付けEEPROM） | SAVE 1 | 0.8.0 |
| RENUM | プログラムの行数を10から10刻みにする(GOTO/GOSUBの飛び先は手で変更必要) | RENUM | 0.6 |
| END | プログラムを終了する | END | |
| REM | これ以降の命令を実行しない（コメント機能） | REM START | |
| BEEP 数,数 | BEEPを鳴らす 周期(1-255)と長さ(1/60秒単位)は省略可 ※EX2-GNDに圧電サウンダー接続 | BEEP | 0.9.0 |
| PLAY MML | MMLで記述した音楽を再生する ※EX2-GNDに圧電サウンダー接続 | PLAY "CDE2CDE2" | 0.9.0 |
| TEMPO 数 | 再生中の音楽のテンポを変更する | TEMPO 1200 | 0.9.0 |

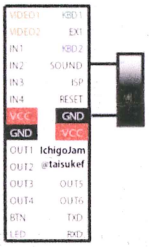
### 上級コマンド

| コマンド | 解説 | 例 | 対応 |
|---|---|---|---|
| [数] | 配列（[0]から[100]までの101コの連続した変数として使えます） LET[0],1,2,3で連続代入可能(ver0.9.4) | [3]=1 | 0.7.6 |
| GOSUB 行番号 / RETURN | 次に実行する行を決め、RETURNで戻ってくる ver0.9.7以降は変数でもOK | GOSUB 100 | |
| LIST 行番号1,行番号2 | 行番号1以上、行番号2以下のプログラムを表示する（行番号1を省略可能） | LIST 10,300 | |
| FREE() | プログラムの残りメモリ数を返す | ? FREE() | 0.8.9 |
| OUT 数 | 外部出力OUT1-6への出力を0-63で指定する（出力は3.3V / 5,6は0.9.0以降） | OUT A | 0.6 |
| IN() | 外部入力IN1-4の状態を0-15の数値で返す（IN3のみプルダウン抵抗なし） | A=IN() | 0.6 |
| 数 & 数 | 論理積（ビットマスク） | ? 3&1 | 0.7.2 |
| 数 \| 数 | 論理和 | ? 3\|1 | 0.8.3 |
| 数 ^ 数 | 排他的論理和（ビット反転） | ? A^1 | 0.7.2 |
| 数 >> 数 | 右シフトする | ? A>>1 | 0.7.2 |
| 数 << 数 | 左シフトする | ? A<<1 | 0.7.2 |
| LRUN (数) | プログラムを読み込み後、実行します | LRUN 1 | 0.8.0 |
| VIDEO 数 | 画面表示、停止を切り替える 0で停止 F8で表示 | VIDEO 0 | 0.9.4 |
| SLEEP | プログラムを休止する(ボタンが押されると復帰して最初から実行) | SLEEP | 0.7 |
| BPS 数 | シリアル通信速度を変更する(0で初期値の115,200bps) | BPS 9600 | 0.7.7 |

CC BY http://ichigojam.net/

## IchigoJam サウンド 拡張リファレンス

実体配線図




IchigoJam の SOUND(または EX2)ピンと GND ピンに、圧電サウンダー、スピーカー、音声端子を接続すると、PLAY/BEEP/TEMPO で音を操ることができます。MML（ミュージックマクロランゲージ）を使った音楽や効果音でプログラムが豊かにしてみましょう。

| MML | 説明 | 例 |
|---|---|---|
| [音] | 音(ドレミファソラシ=CDEFGAB)を鳴らす | PLAY"CDEFG" |
| [音][N] | 長さを指定して音を鳴らす(1/2/3/4/8/16/32) | PLAY"C4E2D1" |
| [音][N]. | 符点音符、長さが指定の 1.5 倍になる | PLAY"C4.E2.D1." |
| [音]+ | 半音上げる | PLAY"CC+" |
| [音]- | 半音下げる | PLAY"DD-" |
| R | 休符（長さ指定、符点にも対応） | PLAY"CRDRE" |
| T[N] | テンポ指定（初期値 120） | PLAY"T60CDE" |
| L[N] | 長さ指定しないときの長さ（初期値 4） | PLAY"L16CCC" |
| O[N] | オクターブで指定（初期値 4）O1C～O5D | PLAY"O1CO5D" |
| > | オクターブを 1 上げる | PLAY"C>C" |
| < | オクターブを 1 下げる | PLAY"C<C" |
| N[N] | 音の高さ指定(1～255) 長さは L で指定 | PLAY"N1N2N4N8" |
| $ | これ以降の MML を繰り返す | PLAY"CDE$GC" |
| | 音を止める（何も鳴らさない） | PLAY |

※　画面出力の水平同期信号を使っているため正確な平均律では鳴りません

CC BY http://ichigojam.net/